SPECIFICATION

TO WHOM IT MAY CONCERN:

BE IT KNOWN that ERIC R. UNER is a citizen of the United States and is a resident of Carpentersville, Illinois, U.S.A. and has invented new and useful improvements in a

AN EMBEDDED WEB SERVER CAPABLE OF MANAGING DYNAMIC CONTENT

DELIVERY OF DATA STREAM, AUDIO STREAM, OR VIDEO STREAM

and do hereby declare that the following is a full, clear and exact description, reference being had to the accompanying drawings and to the numerals of reference marked thereon, which form a part of this specification.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to embedded web server technology. More specifically, my invention is primarily intended for an embedded web server capable of managing dynamic content delivery of data stream, audio stream, or video stream.

Description of the Prior Art

Specialized computers known as "server appliances" are quickly replacing complex all-purpose machines as web servers. The business paradigm is from Cisco Systems Inc., who revolutionized communications with the router, a specialized computer that moves data around networks. Network Appliance, Inc. is a founder of so-called server appliances. The traditional general-purpose servers are built to handle dozens of tasks at once. However, server appliances are built to perform one job quickly and simply. A server appliance designed for web hosting will not crunch numbers as fast as a big general-purpose machine. However, that streamlined web-hosting server can dish out web pages faster than traditional servers costing three times as much. Also, storage servers allow users anywhere on a network to store and retrieve files faster than traditional devices; caching appliances keep popular web pages in close proximity to web servers, which can cut download time by 80%.

A monumental shift in the way technology is delivered is driving the popularity of these server appliances. Instead of using massive computers in their offices, corporations can now buy computing power from other companies via the web. These companies house the computers and the problems, dishing up applications software as its needed to perform specific tasks. But for this new technology-on-tap to work, the web must be as reliable as the phone system. That will fuel demand for single-purpose devices that can handle tasks without the breakdowns or sluggish performance that plague multifunction computers. By incorporating these server appliances to big, general-purpose machines, the overall performance of online networks can be improved significantly.

Various inventions have been made to provide an embedded web server to the server market. United States Patent No. 5,956,487, which issued to Venkatraman, et al., discloses an Embedding Web Access Mechanism in an Appliance for User Interface Functions Including a Web Server and Web Browser. This invention provides a solution for widely accessible, low cost, and enhanced user interface functions for a device. The solution involves embedding web access functionality into the device including a web server that provides a device web page. The device includes an embedded network interface that enables access to the device web page by a web browser. A user of the web browser accesses the user interface functions for the device through the device web page. The web server functionality may be implemented with existing circuitry in a device, such as an exiting processor, memory, and input/output circuitry that normally perform device-specific functions, thereby avoiding the extra cost and space required for dedicated web server hardware for the device. The web server functionality embedded in the device enables the device user interface access via a variety of communication mechanisms including the world wide web portion of the Internet. The costs of providing screen based user interface mechanisms are exported away from the device and do not require an external computer to provide a device web server. The methods and mechanisms disclosed herein provide screen based user-friendly interfaces to a wide variety of devices without the necessity to develop expensive hardware and software applications for differing devices. The methods and mechanisms employ web technology so that access to a device user interface is independent of the computer system platform employed, independent of the web browser software executed, and independent of the location of the user. The user interface information is packaged using he Hyper-Text Markup Language (HTML) and is transported according to the Hyper-Text Transfer Protocol (HTTP). The HTML and HTTP protocols enable communication with existing web browsers independent of the platform that executes the web browser. The present techniques avoid the need of an industry-wide Application Programming Interface (API) to unify the control and use of equipment.

United States Patent No. 5,973,696, which issued to Agranat, et al., discloses an Embedded Web Server. The invention provides an improved graphical user interface (GUI) for use in connection with remote control, management, configuration, monitoring and diagnosing functions embedded in applications, devices and equipment. The invention also provides a method for providing a graphical user interface having dynamic elements. The method begins by defining elements of the graphical user interface in at least one text document written in a mark-up language. Next, the method defines including at a location in the document, a code tag containing a segment of application source code. The text document is then served to a client, which interprets the mark-up language; and when the location is encountered, the client is served a sequence of characters derived from a result of executing a sequence of instructions represented by the segment of application source code. An embodiment of code tags illustrating their use is later described in detail herein. The invention further provides another method for providing a graphical user interface having dynamic elements. This method also defines elements of the graphical user interface in at least one text document written in a mark-up language. Included in the document is a string identified by prototype tags. The text document is served to a prototyping client which interprets the mark-up language but does not recognize and does not display the prototype tag, but does display the string. An embodiment of prototype tags illustrating their use is described in detail hereafter. The invention still further provides another method for providing a graphical user interface having dynamic elements. Elements of the graphical user interface are defined in at least one text document written in a mark-up language. Included at a location in the document is a code tag containing a segment of application source code. Also included in the document is a string identified by prototype tags. The text document is compiled into a content source, which is subsequently decompiled into a replica of the text document. The replica of the text document is served to a client which interprets the mark-up language; when the location is encountered in the replica, the client is served a character stream generated by executing the segment of application source code. The invention further provides a software product recorded on a medium. The software product includes a mark-up language compiler which can compile a mark-up language document into a data structure in a native application programming language, the compiler recognizing one or more code tags which designate included text as a segment of application source code to be saved in a file for compilation by a compiler of the native application programming language. The invention still further provides a method for providing a graphical user interface having displayed forms for entry of data. The steps of this method include defining elements of the graphical user interface in at least one text document written in a mark-up language; naming in the document a data item requested of a user and used by an application written in a native application programming language; and compiling the text document into a content source including a data structure definition in the native application programming language for the named data item. The invention may be practiced in a computer-based apparatus for developing a graphical user interface for an application, the apparatus including an editor which can manipulate a document written in a mark-up language and a viewer which can display a document written in the mark-up language. The apparatus further includes a mark-up language compiler which recognizes a code tag containing a source code fragment in a native application source code language, the code tag not otherwise part of the mark-up language, the compiler producing as an output a representation in the native application source code language of the document, including a copy of the source code fragment.

United States Patent No. 6,170,007, which issued to Venkatraman, et al., discloses an Embedding Web Access Functionality into a Device for User Interface Functions. Web access functionality is embedded in a device to enable low cost widely accessible and enhanced user interface functions for the device. A web server in the device provides access to the user interface functions for the device through a device web page. A network interface in the device enables access to the web page by a web browser such that a user of the web browser accesses the user interface functions for the device through the web page.

Although these patents disclose different embedded web servers, none of the patents teach how to provide an embedded web server with high performance and running without an operating system. The current invention is an embedded web server capable of managing dynamic content delivery of data stream, audio stream, or video stream. The current invention is a high performance, secured, and embedded web server.

The business world is now experiencing a big shift from traditional ways of business commerce to web-based transactions conducted in real-time electronic marketplaces between buyers and sellers. Technology is becoming more important to companies of all sizes and in all industries, and there has been an increased drive to implement Internet solutions. The trends point to a new era in outsourced solutions where there is greater acceptance of standardized critical functions across multiple companies and whole industries. This will accelerate the speed to market and the ability to achieve economies of scale. Deregulation, globalization, and technological innovation are accelerating the rate of change in business.

Organizations face constant pressures to improve the quality of their products and services, reduce cost and time to market, enhance operating efficiencies, and strengthen customer relationships. These pressures are compelling business managers to streamline their processes and improve the flow of information both internally and externally. Businesses are spending more money on infrastructure. Explosive volume gains and the drive among enterprises of all classes to implement Internet and business solutions are the prime factors fueling the growth of the Internet infrastructure.

As businesses implement their Internet, a higher level of quality of service and reliability will be expected. This demand for quality should further drive the demand for technology products and services, with infrastructure being among the most critical components of this spending. For this to work, all information systems of an enterprise must be integrated to enable information to pass seamlessly across the enterprise.

Due to the increased need for better connectivity, more servers, and more Internet-enabling devices, there has been huge growth in more specialized devices upon which numerous technologies are converging. Many of these involved multimedia and Internet connectivity. There has been heightened interest in the evaluation of embedded Internet products and technology worldwide. As increasingly more dynamic content is demanded over the Internet, enterprises with intensive online business will need high-performing, reliable, scalable, and secure web servers and services to support their business.

As a result of all these needs, what is needed then is a secured embedded web server capable of managing dynamic content delivery of data stream, audio stream, or video stream.

Accordingly, it is a principal object of my invention to provide an embedded web server that is capable of managing dynamic content delivery of data stream, audio stream, or video stream.

It is a further object of my invention to provide an embedded web server that is secured and hacker proof.

It is a further object of my invention to provide an embedded web server that is capable of running high-performance electronic commerce web sites.

It is a still further object of my invention to provide an embedded web server that is capable of embedding and securing data, content, protocols and scripts.

It is a further object of my invention to provide an embedded web server that is capable of delivering real time response.

It is a further object of my invention to provide an embedded web server that is capable of operating without an operating system

It is a further object of my invention to provide an embedded web server that is capable of running without human intervention once an initial configuration is completed.

SUMMARY OF THE INVENTION

According to my present invention I have provided an embedded web server capable of managing dynamic content delivery of data stream, audio stream, or video stream. The embedded web server comprises a microprocessor, a memory, a network port, a dynamic internet streaming engine, a security key generating application capable of generating security keys, and an object-oriented language. The embedded web server is capable of functioning as a web server. The embedded web server is capable of running high-performance electronic commerce web sites. The embedded web is also capable of embedding and securing data, content, protocols and scripts. The embedded web server is further capable of delivering real time response. The embedded web server is capable of operating without an operating system. The embedded web server is capable of creating a template. The template is then compiled and loaded into said embedded web server. The template is capable of specifying protocol, content, data, and being a scripting language to translate inbound requests. As soon as the embedded web server is completed for the initial configuration, it is capable of running without human intervention. The dynamic internet streaming engine of the embedded web server is specifically designed to facilitate rapid application development and web site prototyping. The dynamic internet streaming engine includes IP level security features. The dynamic internet streaming engine also allows developers to quickly add or change features without the need for code changes. The dynamic internet streaming engine is capable of enabling said embedded web server to deliver dynamic text and binary data stream editing for performing complex actions and making interactive web. The dynamic internet streaming engine is also capable of propagating changes made in one place instantly to thousands of files. The dynamic internet streaming engine is further capable of compiling code directly into HTML. The dynamic internet streaming engine is capable of facilitating rapid application development and web site prototyping. The security key generating application of the embedded web server is capable of generating security keys based on both a mathematical and biological equation. The security keys generated by the security key generating application are guaranteed to be unique and random. The object-oriented language used by the embedded web server allows access to preprocess directives from both C++ code and HTML code.

DESCRIPTION OF THE DRAWINGS

Other objects of my invention, as well as particular features, elements, and advantages thereof, will be elucidated in, or apparent from, the following description and the accompanying drawing figures.

Other features of my invention will become more evident from a consideration of the following detailed description of my patent drawings, as follows:

Figure 1 is a preferred embodiment of an embedded web server;

Figure 2 is a flow diagram of the operating process of the embedded web server;

Figure 3 illustrates of the security sub system;

Figure 4 shows functional structure of the embedded web server;

Figure 5 demonstrates the operational relationship between the real time executor and other part of the embedded web server;

Figure 6 portrays the operational relationship between the board support package and other part of the embedded web server;

Figure 7 illustrates the operational relationship between the embedded memory manager and other part of the embedded web server;

Figure 8 discloses the operational relationship between the object-oriented language and other part of the embedded web server; and

Figure 9 further discloses the operational relationship between the device interface and other part of the embedded web server.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 is a preferred embodiment of an embedded web server 10 capable of managing dynamic content delivery of data stream, audio stream, or video stream. Said embedded web server 10 comprises a microprocessor 11, a memory 12, a network port 13, a dynamic internet streaming engine 14, a security key generating application 15 and an object-oriented language 16.

The dynamic internet streaming engine 14 is capable of enabling said embedded web server 10 to deliver dynamic text and binary data stream editing for performing complex actions and making interactive web. Propagating changes made in one place by the dynamic internet streaming engine 14 will be instantly passed over to thousands of files. The dynamic internet streaming engine 14 is capable of compiling code directly into HTML, and facilitating rapid application development and web site prototyping.

The security key generating application 15 is capable of generating security keys based on both a mathematical and biological equation. The mathematical and biological equation can guarantee the uniqueness and randomness of the security.

The object-oriented language 16 used by said embedded web server 10 allows access to preprocess directives from both C++ code written by a developer and HTML code written by a graphic artist or web designer. Therefore, it will make the communication between web developer and web designer or a graphic artist very easy and effective.

The embedded web server 10 can be used as a static page web server, a dynamic page web server, a random session ID or sequence number generator server, a caching device, a secure server (https server), a web server, a content acceleration server, an internet standard file transfer protocol (FTP) server. The embedded web server 10 can be used to run E-Commerce applications, such as data warehousing, credit card transaction processing, etc.

The embedded web server 10 is capable of running high-performance electronic commerce web sites, embedding and securing data, content, protocols and scripts, delivering real time response, functioning as a web server, operating without an operating system and running without human intervention once an initial configuration is completed.

The embedded web server 10 is capable of creating a template, which is then compiled and loaded into said embedded web server 10. Said template is capable of specifying protocol, content, data, and being a scripting language to translate inbound requests.

The dynamic internet streaming engine 14 is specifically designed to facilitate rapid application development and web site prototyping. The dynamic internet streaming engine 14 includes IP level security features. The dynamic internet streaming engine 14 allows developers to quickly add or change features without the need for code changes.

Referring now to Figure 2, which is a flow diagram of a operating process of the embedded web server 10. The operating process starts at block 21 and proceeds to block 22, which is a first network interface 32. The first network interface 32 is the interface for all the incoming data to go into embedded web server 10. The operating process proceeds to block 23, which is the network port 13, where the incoming data enters the embedded web server 24. The operating process proceeds to block 24, where the data is processed by the embedded web server 10. The operating process proceeds to block 25, which is a security sub-system 31. The security sub-system 31 runs a security key generating application 15 to generate session identifiers that are random and never repeating. The operating process proceeds to block 26, which represents other subsystems required by the process, where it will perform all the functions of a web server, but at a high speed, and generate outgoing data. Block 26 may include various kinds of subsystems, including FTP, HTTP static pages or dynamic pages, security sub-system similar to the security sub-system 31, or the like. The operating process then proceeds to block 27, where the outgoing data goes out through a second network port 13. The operating process then proceeds to block 28, where the outgoing data goes back to a second network interface 33. The operating process proceeds to block 29 and ends right there. The first network port 13 and the second network port 13 can be the same network port 13. The first network interface 32 and the second network interface 33 can be the same network interface 33.

Referring to Figure 3, which shows the details of the security sub system 31 to generate session identifiers. The security sub system 31 is a means for producing a session identifier 40. The means for producing said session identifier 40 is capable of generating said session identifiers, which are never repeating within a certain period of time and difficult to guess. The means for producing said session identifier 40 comprises a character-generating application 41 and an external timer device 51. The external timer device 51 connects to the character-generating application 41 and is capable of providing both a current time and a periodic tick to the character-generating application 41. The character-generating application 41 exists as a single task in the embedded web server 10.

The character-generating application 41 comprises a character generator 42, a random generator 43 connecting to the character generator 42, a temporal reference storage 44 connecting to the character generator, the temporal reference storage 44 storing the most current time information, and a key-pool group 45 connecting to the character generator, the key-pool group 45 including any or all of a first kind of key with a first kind of pool, a second kind of key with a second kind of pool, and a third kind of key with a third kind of pool. The character generator 42 is capable of generating a character set, the character set having thirty-two different characters, based on a geometric progression of x(n)=p(x(n-1) + i), the geometric progression manifesting itself as a chaotic progression of orbits around an origin, the orbit being defined as a unique and continuous path around the origin and never crossing in on itself or any other orbit, the x(n) and the x(n-1) representing different character sets, the n representing the number of the character sets generated by the character-generating server, the i representing a temporal difference between the time when two sequential orbits cross an arbitrary infinite vector from the origin, the p representing a period, the period being the temporal difference between character sets along any of the orbits, the geometric progression defining thirty-two periods on any of the orbit, the character set being a first kind of character set, a second kind of character set or a third kind of character set.

Referring now to Figure 4, which is a functional structure of the embedded web server 10. The embedded web server 10 has seven functional parts, including a real time executor 52, subsystem 53, board support package 54, device interface 55, an object-oriented language 56, embedded memory manager 57, and information storage interface 58. Figures 5-9 demonstrates how these functional parts are related to each other.

Referring now to Figure 5, which demonstrates the operational relationship between the real time executor 52 and four of the other six functional parts, including subsystem 53, board support package 54, device interface 55, and object-oriented language 56. During the operation of the embedded web server 10, the real time executor 52 might communicate to various subsystems 53 to perform various functions of a web server. The real time executor 52 may also communicate to the board support package 54 to get various support it needs to operate. The real time executor 52 may also communicate to the device interface 55 to communicate to the outside of the embedded web server 10. The real time executor 52 may also use the object-oriented language 56 to perform various functions.

Referring now to Figure 6, which demonstrates the operational relationship between the board support package 54 and four of the other six functional parts, including subsystem 53, embedded memory manager 57, device interface 55, and object-oriented language 56. During the operation of the embedded web server 10, the board support package 54 might communicate to various subsystem 53 back and force to support the need of various subsystem 53. The board support package 54 might also communicate to the embedded memory manager 57 to support its function. The board support package 54 might also communicate to device interface 55 to support its operation. The board support package 54 might also communicate with the object-oriented language during the operation of the embedded web server 10.

Referring now to Figure 7, which demonstrates the operational relationship between the embedded memory manager 57 and four of the other six functional parts, including subsystem 53, board support package 54, device interface 55 and the object-oriented language 56. During the operation of the embedded web server 10, the embedded memory manager 57 communicates to various subsystem 53, board support package 54, device interface 55 and the object-oriented language 56 to provide the necessary memory management function.

Referring now to Figure 8, which demonstrates the operational relationship between the object-oriented language 56 and four of the other six functional parts. The six functional parts are the subsystem 53, the embedded memory manager 57, the real time executor 52 and the information storage interface 58. During the operation of the embedded web server 10, the object-oriented language 56 provides all the language function and support for the needs from the subsystem 53, embedded memory manager 57, real time executor 52 and information storage interface 58.

Referring now to Figure 9, which demonstrates the operational relationship between the device interface 55 and five of the other six functional parts. The six functional parts are the subsystem 53, the embedded memory manager 57, the real time executor 52, the information storage interface 58 and the object-oriented language 56. The device interface 55 communicates to the object-oriented language 56 through the information storage interface 58. The device interface 55 also communicates to the subsystem 53, embedded memory manager 57, and the real time executor 52 during the operation of the embedded web server 10.

Different functional parts of the embedded web server 10 are integrated into the embedded web server 10 to make the embedded web server 10 more powerful and efficient than a web servers.

Hence, my current invention provides an embedded web server that is capable of managing dynamic content delivery of data stream, audio stream, or video stream.

My current invention also provides the following new and improved features:

1. an embedded web server that is secured and hacker proof;

2. an embedded web server that is capable of running high-performance electronic commerce web sites;

3. an embedded web server that is capable of embedding and securing data, content, protocols and scripts;

4. an embedded web server that is capable of delivering real time response;

5. an embedded web server that is capable of operating without an operating system; and

6. an embedded web server that is capable of running without human intervention once an initial configuration is completed.

As various possible embodiments may be made in the above invention for use for different purposes and as various changes might be made in the embodiments and methods above set forth, it is understood that all of the above matters here set forth or shown in the previously described accompanying drawings are to be interpreted as illustrative and not in a limiting sense.